

An Efficient Approach for Scheduling Imaging Tasks Across a Fleet of Satellites

James Boerkoel,¹ James Mason,² Daniel Wang,² Steve Chien,² Adrien Maillard²

¹ Harvey Mudd College

² Jet Propulsion Laboratory, California Institute of Technology
boerkoel@hmc.edu, firstname.lastname@jpl.nasa.gov

Abstract

Dynamically retasking satellites in response to scientific alerts is challenging because the tasks and opportunities of one satellite can influence another. This abstract focuses on our high-level approach for scheduling imaging tasks across a constellation of satellites, which is subject to orbital and other practical constraints such as finding a feasible up-/down- link schedule. Our approach is inspired by combining insights from two existing approaches about the structure of these problems to create an efficient, new approach. We show that our approach stacks up favorably against two baselines—an optimal solver as well as a naive, greedy approach.

Introduction

The shift towards small sats and cube sats has exploded the size of remote sensing constellations. Planet and Spire both maintain constellations of over 100 spacecraft, with plans to expand. Many other organizations are following with networks of their own. These large constellations may require new scheduling techniques to improve operations.

We are working on developing new observing systems to handle these increased capabilities along with direct integration with science models. These science models can identify or project events of interest, such as flooding or a volcano eruption, which can be used to trigger additional observations by target-able sensors (Chien et al. 2020).

This work focuses on the subproblem of imaging task scheduling of point targets. We take scientific alerts and feed them into the scheduler. To make the problem more realistic, we also give the scheduler contending targets. These targets consist of a set of volcanoes, various large cities, and random target points around the Earth’s landmass. We plan for several satellites, including the Planet Skysat constellation as well as various NASA satellite assets capable of imaging.

Related Approaches

There have been many recent approaches proposed for solving the imaging and downlink satellite scheduling problem (Augenstein et al. 2016; Nag, Li, and Merrick 2018; Shah et al. 2019). Here, we briefly highlight two approaches that form the inspiration for our approach.

Copyright ©2021 All rights reserved.

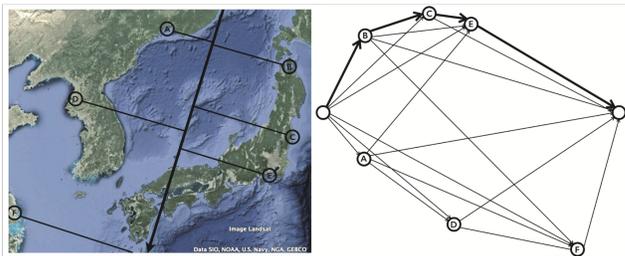


Figure 1: Image opportunities over the Sea of Japan (left) translated into a DAG (right).

Credit: Augenstein et al. (2016), Fig. 2 & 3

Augenstein et al. (2016) formulates a Mixed Integer Linear Program (MILP) for simultaneously solving both the imaging and downlink problem for a constellation of roughly 12 satellites. The intractability of this MILP leads to an approach to solving the downlink and imaging problems sequentially. We omit the details of how the problem is decoupled into two sequential subproblems and focus instead on their approach for optimally solving each local satellite’s imaging scheduling problem. This can be done efficiently by representing opportunities as a Directed Acyclic Graph (DAG), where directed edges represent feasible transitions between opportunities, and then finding the maximally weighted path through the DAG using Dynamic Programming (DP). The solutions are guaranteed optimal primarily because each satellite’s image scheduling problem is solved independently, and an additive objective function is used—our approach relaxes both of these requirements. The results from these DPs are used to provide the MILP estimates of the amount the cumulative payload and priorities of the imagery that a satellite might obtain between downlink opportunities.

Nag, Li, and Merrick (2018) focuses on solving the global image schedule optimization across a constellation of CubeSats. They provide both a globally optimal, but intractible MILP for solving the problem, and also propose a way to decompose the problem into a set of DPs for solving each satellite’s problem locally optimally.

The DP approach described creates a table of size $t \times p$, where t is a discretization of time up to a horizon and p is

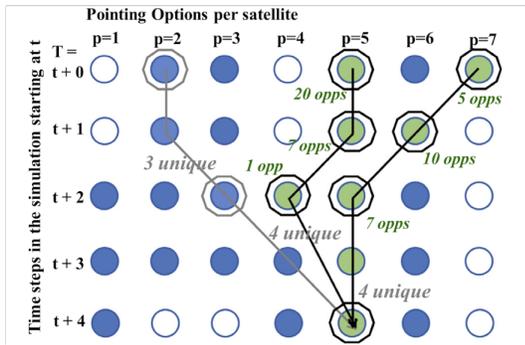


Figure 2: Cartoon illustration of DP algorithm. Credit Nag, Li, and Merrick (2018), Fig. 11.

a set of unique target positions. Then, the DP table maintains the invariant that cell i, j contains the set of optimal sequence of unique images that a satellite pointing at i at time j could have collected. However, Nag, Li, and Merrick (2018) points out that this sequence is unlikely to be unique, and so propose heuristics for maintaining a small set of sequences that achieve the optimal priorities.

While each satellite’s DP can be solved efficiently, the global situation suffers from a lack of information sharing, leading to duplicates images across satellites. Maintaining this information across DP tables would lead to a combinatorial explosion. So instead, Nag, Li, and Merrick (2018) propose breaking the problem into very short time horizons, where within each time horizon each satellite’s subproblem is solved independently. Then, after each of these short horizons have passed, satellites exchange information about which targets they have imaged.

A Sparse DAG Approach to Solving the Local Image Scheduling Problem

Our approach to scheduling satellite imaging blends key insights from both Augenstein et al. (2016) and Nag, Li, and Merrick (2018). Both approximation approaches essentially boil down to dynamic programs. The advantage of Augenstein et al. (2016) over Nag, Li, and Merrick (2018) is that it does not require a discretization of time, thus significantly reducing the number of entries in the DP-table representation. So, whereas the size of Nag, Li, and Merrick (2018)’s is $O(tp)$, where t is number of time units represented and p is the number of pointing positions, the size of Augenstein et al. (2016)’s DP table is $O(n)$, where n is the number of opportunities / nodes in the DAG. While p can be treated as a constant, when n is much less than t , we would expect a table of size $O(n)$ to be much more compact.

The advantage of Nag, Li, and Merrick (2018) over Augenstein et al. (2016), on the other hand, is that each cell requires visiting only a fixed number p of previous cells in order to update, whereas Augenstein et al. (2016) has to consider all possible $O(n)$ previous opportunities. Thus, the time to fill a cell in the DP table is constant vs. linear.

Our approach attempts to blend the benefits of these two approaches together. Like Augenstein et al. (2016), we rep-

resent the problem as a DAG, where each imaging opportunity is represented as a node and where edges connect imaging opportunities that are consistent with one another. Further, a directed edge is added from one node to another if there is sufficient time for the satellite to slew from the first opportunity to the second. This edge is given a weight that is equal to the utility of the second opportunity. A node’s utility can be expressed as a function of the priority of the target and the quality of the opportunity. Once this DAG is established, the local problem can be solved using a DP approach similar to both previous approaches. This is guaranteed to return an optimal solution when the overall objective is the sum of utilities of all opportunities.

A key insight in improving the efficiency of this approach is that the structure of the problem can be exploited to ensure a sparse graph. Each node must be connected to at least the first consistent node that follows. However our formulation does not add edges for nodes that extend beyond that node more than a horizon that is equal to the maximum possible slew time. This assures that the maximum in-/out- degree of any node is bound by $O(p)$, instead of $O(n)$. Thus, the DAG would have $O(np)$ edges instead of $O(n^2)$, and each node would require visiting $O(p)$ neighbors. In short, our DP-based approach is inspired by both previous approaches, but exploits the temporal-spatial structure of the problem to create a DAG with n nodes, where each node takes $O(p)$ to update, for a total runtime complexity of $O(np)$, as compared to $O(tp^2)$ and $O(n^2)$ of previous approaches.

Of course, many objective functions do not follow a general additive independence assumption. For instance, if the goal is to maximize coverage with the best quality possible, then including multiple images of the same target may not improve the objective. For these more general, black-box objective functions, a longest path approach is no longer guaranteed to return an optimal solution. Thus, maintaining a single, best-path according to a black box objective function is a heuristic approach. However, we are able to show empirically that loss in utility is negligible compared to the gains in efficiency over an exact, MILP-based formulation of the problem. In this paper, we focus on the maximal coverage problem—the problem of covering as many targets at the highest utility possible.

Forward Sweep (FS)

The simplicity of our approach lends itself to relatively straightforward opportunities for heuristically improving the solution. First, we explain why our longest-path, DAG-based approach may return a path that omits opportunities due to the myopic nature of the algorithm—when determining which path is best, only the backwards, and not the potential future, utility is considered. So, a returned path might have chosen opportunity A, which targets location X, over B, which targets location Y because at the time it raised the overall objective more. However, later, an opportunity C which also targets X but with a higher utility function is chosen, making A superfluous. Thus, in hindsight, B could have been included in the original path.

A simple, greedy approach to mitigating these examples of opportunity cost is to first scan through all currently com-

puted paths and look for opportunities that are superfluous, and thus can be removed from the path. Next, we scan through all non-scheduled opportunities and check whether each opportunity can be consistently added to and improves the overall utility of the current set of satellite paths.

If an opportunity for improving the overall utility is found, it opens up the possibility that an existing opportunity to image the same target can now be safely removed from a satellite’s path, bringing potentially new opportunities to greedily improve the solution. Thus, this forward sweep procedure can be applied until convergence is reached—if no new additions are found. The efficiency of this greedy forward sweep can be improved by tracking and only considering the set of non-superfluous unscheduled opportunities.

Iterative Improvement (II)

We handle the coordination of satellites by re-solving the satellite’s local problems multiple times, each time taking into consideration the solutions from other satellites’ previous round of computation. This results in an overall iterative improvement of the global objective until some convergence threshold is met. During the iterative improvement process, we can take advantage of cached information from previous solves to improve efficiency. Information can similarly be shared from one satellite’s problem to the next *within* a single round of computation to improve efficiency.

The information from one satellite’s local can seed the subsequent satellites local computation. For instance, if the first satellite captures an image of a target X with utility Y , then the second satellite only needs to consider opportunities that also target X if they have utility that is higher than Y , pruning the rest. Further, since the second satellite knows that the first satellite already plans to acquire an image of X with utility Y , we can update the utility of any opportunities that target X within the DAG of the second to be the *gain* in utility over Y , allowing the second satellite to make a more informed calculation about trade-offs in opportunities. Finally, during subsequent iterations, the efficiency of the longest path (re-)computation can be improved by skipping over any opportunities at the beginning of the satellites path that still represent the best known opportunity for a given target and only begin recomputing once the possibility of a delta has been detected.

Overflight Imaging Benchmark Creation

In this section, we briefly describe our process for creating realistic benchmarks for tasking a constellation of satellite with imaging targets of interest.

Target Benchmarks

To emulate tasking satellites with acquiring images of scientifically interesting phenomenon, we gathered volcanic alerts using MODVOLC¹, a tool that uses NASA’s MODIS instrument to monitor the Earth’s surface for thermal evidence of volcanic activity. We gathered the list of all alerts across a 1-week period across 52 independent weeks of a 1 year period forming 52 unique sets of targets.

¹<http://modis.higp.hawaii.edu/>

We complement these image targets of high scientific value with additional targets of interest to mimic the fact that satellites are often tasked with additional requests. The first source of these additional targets was a list of 572 large cities across the Earth. When combined with the volcanic targets, this forms 52 sets of imaging targets each with just over 600 targets. Finally, to form an even larger set of targets, we randomly sample from a list of 67,000 equidistant coordinates that cover the entire landmass of the Earth until the total number of targets reaches 4000.

Satellites

We emulate tasking a constellation of 15 commercially available assets from Planet’s Skysat constellation, combined with seven NASA and ESA Earth-observing satellites (Terra, Aqua, Suomi NPP, Sentinel-2A, Sentinel-2B, Sentinel-3a, Sentinel-5P). While the later assets are not taskable in real life, their orbits were used to add variety.

Overflight Calculation

To generate satellite overflight data, we model the orbits of satellites using publicly available Two-Line Element (TLE) files. The quality of this model decreases as time moves further from the time of the TLE’s creation, but TLEs are updated frequently based on each satellite’s true position and orbital mechanics. We assume each satellite has a nadir-pointing, fixed field-of-view sensor.

Based on these orbital models, we calculate distance between the satellite, target pairs over the considered time interval, sampling at a fairly coarse level (on the order of 10 minutes). This generates an array of distances between satellite and target. We find the local minima in this array to create a set of coarse estimates of overflight times. From those coarse estimates, we perform a bounded optimization around each minima to find a more precise overflight time, allowing time to vary by the sampling interval at most. For each potential overflight, we then calculate the field-of-view angle required for the satellite to image the target. If the target satisfies field-of-view constraints, we add it to the list of overflights for that satellite, target pair. We adjust the objective corresponding to capturing the target with that satellite based to mimic the fact that the quality of images can degrade when taken off-nadir,

Empirical Analysis

We test the efficacy and relative trade-offs of our proposed approach by comparing it against two other approaches. First is an approach that should compute the exact solution using the MILP characterization of the problem described by Augenstein et al. (2016) and using the IBM ILOG CPLEX Optimization Studio (CPLEX) (ILOG 2019). Second is a greedy approach that uses a variant of the Forward Sweep procedure described above, but does so in sorted order by opportunity utility. We compare that against three versions of our above approach:

- **DAG** - our DAG approach described above without the iterative improvement of forward sweep heuristics applied;

- **DAG+FS** - the DAG approach with the forward sweep post processing applied, but not iterative improvement;
- **DAG+II+FS** - the DAG approach with both the forward sweep and iterative improvement.

We ran this across two benchmarks. The first benchmark included 52 sets of roughly 600 targets each, which include the volcanic targets of interest and list of large cities as described, and computed overflight information for an entire week’s worth of orbits. The second benchmark included 52 sets of roughly 4000 targets each, which include the volcanic, cities, and landmass targets as described above, and spanned a single 24-hour period worth of overflights. The averages across all 52 weeks for each of these benchmarks are reported in Tables 1 and 2 respectively. The utility of imaging opportunities were adjusted to account for being off-nadir and also for staleness (with a preference for images that could be taken sooner rather than later).

	Solve Time (seconds)	Speedup	# of targets (relative to optimal)	obj. val.
CPLEX	20.29	–	–	–
Greedy FS	2.91	6.97	99.54%	90.87%
DAG	0.84	24.08	99.71%	99.76%
DAG+FS	1.19	17.10	99.71%	99.97%
DAG+II+FS	2.91	6.98	100.00%	99.92%

Table 1: Results across a small set of targets (~600 targets, volcanic+cities) for 7-day period.

	Solve Time (seconds)	Speedup	# of targets (relative to optimal)	obj. val.
CPLEX	37.29	–	–	–
Greedy FS	36.13	1.03	91.86%	69.16%
DAG	5.81	6.42	99.65%	98.44%
DAG+FS	6.67	5.59	99.96%	99.57%
DAG+II+FS	16.04	2.33	99.94%	99.63%

Table 2: Results across a larger set of targets (~4000 targets, volcanic+cities+landmass) for a 24-hour period.

These results highlight a few trends. Importantly, our approach demonstrated that we can significantly improve solve time while only negligibly impacting the number and overall quality of the images acquired.

The results in Table 1 points to the first benchmark being relatively easy to find a reasonably high level of success. The basic greedy-FS approach captures 99.5% of opportunities that the optimal solution, at a quality of just under 91%. However, our DAG-based approaches achieve these results an order of magnitude more quickly than the exact MILP-based approach returned by CPLEX. The benchmark in Table 2 was a much denser set of opportunities, with greater resource contention. This slowed all approaches down and decreased the relative margin by which our DAG-based approaches reduced the solve time. However, our approaches achieve a high-level of optimality, despite returning an approximate solution.

We suspect that slow performance of the II procedure was due to a relatively long tail of convergence. We hypothesize

that the II idea might achieve better performance if we set a convergence criteria that uses a cost/benefit heuristic rather than waiting to achieve full convergence, which as shown, achieves marginal gains at high computational costs.

The relatively slow solve times of the greedy-FS approach were due to the algorithm being implemented rather naively, with every opportunity being considered for addition to the schedule. Further, since opportunities were considered in order of utility, we have to search for the correct insertion point into the path each time.

Conclusion

Overall, we show empirically our approach that solves the global problem as loosely-coupled DAGs can lead to improved solve times over an exact MILP-based formulation with only modest losses in overall solution quality. In the future, we plan to test our approach on larger, more diverse set of benchmarks and to better characterize the types of problems (structure, contention, etc) where our approach is best suited. We also plan to compare our approach against other recent approximate algorithms for solving the satellite image tasking problem, including the two approaches that inspired our approach. Finally, we hypothesize that our iterative approach has an added benefit over a MILP-based formation in that the iterative improvement easily allows targets to be easily added, removed, or their priorities adjusted within a DAG without requiring a full re-solve and plan to characterize the trade-offs in play.

Acknowledgments

This work was performed at the Jet Propulsion Laboratory, managed by the California Institute of Technology, under contract to the National Aeronautics and Space Administration.

References

- Augenstein, S.; Estanislao, A.; Guere, E.; and Blaes, S. 2016. Optimal scheduling of a constellation of earth-imaging satellites, for maximal data throughput and efficient human management. In *26th International Conference on Automated Planning and Scheduling (ICAPS-16)*, 345–352.
- Chien, S.; Boerkoel, J.; Mason, J.; Wang, D.; Davies, A.; Mueting, J.; Vittaldev, V.; Shah, V.; and Zuleta, I. 2020. Leveraging Space and Ground Assets in A Sensorweb for Scientific Monitoring: Early Results and Opportunities for the Future. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, 3833–3836.
- ILOG, I. 2019. V12.10: User’s Manual for CPLEX. *International Business Machines Corporation*.
- Nag, S.; Li, A. S.; and Merrick, J. H. 2018. Scheduling algorithms for rapid imaging using agile Cubesat constellations. *Advances in Space Research* 61(3): 891–913.
- Shah, V.; Vittaldev, V.; Stepan, L.; and Foster, C. 2019. Scheduling the World’s Largest Earth-Observing Fleet of Medium-Resolution Imaging Satellites. In *Proc. of 11th International Workshop on Planning and Scheduling for Space (IWSS 2019)*.